Apeared in *ECCV '94, Stockholm, May 1994.*

Integration and Control of Reactive Visual Processes[1]

James L. Crowley and Jean Marc Bedrune
IMAG - LIFIA, 46 Ave Félix Viallet, 38031 Grenoble, France

Morten Bekker and Michael Schneider
Lab of Image Analysis, Aalborg University, Fr. Bajers Vej 7, DK-9220 Aalborg , Denmark

Abstract

This paper describes a new approach to the integration and control of continuously operating visual processes. Visual processes are expressed as transformations which map signals from virtual sensors into commands for devices. These transformations define reactive processes which tightly couple perception and action. Such transformations may be used to control robotic devices, including fixation an active binocular head, as well as the to select and control the processes which interpret visual data.

This method takes inspiration from so-called "behavioural" approaches to mobility and manipulation. However, unlike most previous work, we define reactive transformations at the level of virtual sensors and device controllers. This permits a system to integrate a large number of perceptual processes and to dynamically compose sequences of such processes to perform visual tasks. The transition between visual processes is mediated by signals from a supervisory controller as well as signals obtained from perception. This method offers the possibility of constructing vision systems with large numbers of visual abilities in a manner which is both scalable and learnable.

After a review of related work in mobility and manipulation, we adapt the reactive process framework to computer vision. We define reactive visual processes which map information from virtual sensors to device commands. We discuss the selection and control of reactive visual processes to accomplish visual tasks. We then illustrate this approach with a system which detects and fixates on different classes of moving objects.

1. Introduction

As available computing power has increased, it has become possible to build and experiment with vision systems that operate continuously. One result has been a rapid advance in the robustness and sophistication of vision techniques, as complex and fragile techniques have been discredited by experiments and replaced by techniques which stress low computational-complexity and robustness. These advances have brought us to a new aspect of computer vision: the integration of a large number of visual processes into a single system, and the control of attention and processing within such a system.

The framework developed in this paper is the result of several years of experiments in the integration and control of continuously operating integrated vision systems [15]. Initial experiments have involved hand-coding perceptual processes and their control procedures. This early approach proved rigid and difficult to adapt to new tasks. By reformulating our system using the approach described in this paper we have obtained a system in which:

1) It is possible to formally prove properties about compositions of perceptual processes [24].
2) Perceptual processes may be learned using connectionist approaches to function approximation (such as Back-propagation[4], ART [8], radial basis functions), or

[1] Funded by the CEC DG III, ESPRIT Basic Research project EP 7108 Vision as Process II

defined using techniques from signal processing (such Gabor filters or Correlation).

3) Compositions of perceptual processes may be automatically formed using techniques such as reinforcement learning [27], [9], [19], or determined using rule based planning techniques.

This approach is inspired by an approach to mobile robotics often referred to as "Behavioural" [5]. The notion of a reactive behaviour has been shown to provide a compact and general formalism for such tasks as grasping and haptic exploration [26], autonomous vehicle driving [23], and navigation [11].

Criticisms of this “behavioural” approach to robotics contend that

1) Inhibition based control regimes, such as subsumption [5], are inadequate for constructing complex systems,
2) Perception without intermediate representations are subject to an exponential explosion in computational cost.
3) The concept of "goal" is fundamental to systems which must perform useful tasks in changing environment.

Our experience confirms these criticisms. However, reactive transformations may be used with control regimes other than subsumption. Indeed a number of researchers are beginning to look at the use of other forms of composition of primitive reactive transformations [22]. Furthermore, this approach opens the possibility of techniques for learning compositions of robot behaviours [27] through such techniques as reinforcement learning. With regard to the second criticism, it is possible to base the perceptual space on functions computed from internal representations of the world. Such an approach permits the complexity reduction benefits of intermediate representations [28]. Furthermore, goals may be included as intermediate representation.

The framework we propose is related to the work of Kosecka and Bajcsy [20] on the use of state transition networks formalised as a Discrete Event Dynamics Systems (DEDS) notation [24]. In their approach, composite reactive transformations are hand crafted and expressed in a formal tool in order to prove properties about such compositions. We have approached the problem from a view point of obtaining a framework in which the composition of transformations can be controlled by a rule-based planning system. Furthermore, we believe that this approach opens the possibility of acquiring both visual processes and their composition using connectionist approaches to machine learning.

A crucial problem in a continuously operating vision system is dealing with the very large quantity of ambiguous and noisy data provided by cameras. An often overlooked property of the human visual system is that the perceptual processes are serial and highly restrictive about what data is processed at each instant. The human visual system can be seen as a pipeline of filters for eliminating unnecessary information. Even before the visual data arrives at the retina, it is restricted to a narrow depth of field by the optics of the eye. The region of the world perceived is even more severely filtered by simple processes which restrict attention to the horopter (those parts of the world which project to the same location in the stereo retinas). The horopter is moved dynamically around the scene by saccadic movements of the eyes, limiting the perception at each instant to a narrow slice of the world. The primary role of binocular vision thus seem to be separation of figure and ground, and not 3D reconstruction. Active vision systems take inspiration from this "filtering" principle to limit the amount of data which must be attended to in order to provide a response within a fixed delay.

Active vision may be defined as "Control of cameras and control of processing to aid the observation of the world". A number of researchers have provided striking demonstrations of systems which perform simple visual tasks in real time using this principle [2], [3], [6],

[17], [21], [22]. However, in each case the system was limited to demonstrating the advantages which active control brought to a particular visual process. Little has been done on the problem of extending an active approach to all levels of the vision system and adapting such an approach in a system composed of a large number of visual behaviours. This paper presents a framework for such integration and control based on reactive transformations.

2. Reactive Visual Process

In order to place our framework on a solid foundation, this section presents definitions of reactive visual processes and their components. These definitions are then used in section 3 to develop a framework for integration and control of visual processes. Examples of these concepts are presented in section 4.

2.1 Perceptual Spaces

Perceptual systems make observations of the external world through perceptual organs or “transducers”. We define a transducer as an organ which provides a digitized measure of some property of the world from a region of space during an interval of time. The result is a digital signal which may be a scalar, a vector, an image, or even a vector of images. This measured property partially reflects the “state” of the external world. For example, the composition of the lenses, retina, camera electronics and digitizer which provide images to a machine vision system constitutes a transducer. The resulting signal has as many dimensions as pixels in the image.

Brooks has argued [5] that robotic systems can be composed of reactive behaviours which map directly from transducers to actuators. While such an approach is possible, it does not scale well to non-trivial processes. In order to go beyond the purely reactive behaviours of insect-like systems, it is necessary to reduce computational complexity by introducing intermediate processing. This intermediate processing may involve fusing signals acquired at different times to construct an intermediate description (or estimate) of the state of the external world, (a local model). Such an intermediate description can provide input for a large number of visual processes with a minimum of computations.

Let us define an intermediate representations as a collection of properties, $R_i(t)$. Among the intermediate representations, we include such things as the current systems goals and information from long term memory. This provides a way to include the system goals within a reactive visual process. We define a virtual sensor to be a digitized time sampled function, $S_i(t)$, which is computed on a subset of the set of transducers $T_i(t)$ and intermediate representations $R_i(t)$. Examples of virtual sensors include a bank of space-time Gabor filters applied to an image sequence [29], perceptual grouping procedures applied to a gradient image, and the current goal for a mobile robot expressed as a position relative to the robot.

A perceptual space is a vector space defined by a set of virtual sensors. Thus a perception, $P_k(t)$, is a vector in a perceptual space.

$$P_k(t) = \{S_1(t), S_2(t), ... , S_n(t)\}$$

An important role of virtual sensors is to reduce the number of dimensions required for a perceptual space.

A perceptual signal is a signal which is created when a perception occurs within a pre-defined region of a perceptual space. Perceptual signals are used to signal a change in state within a visual process. This change in state may be planned (such as finding a landmark object) or unexpected (an avoidance reflex triggered by motion).

2.2 Action Spaces

A symmetry exists between perception and action. Each of the concepts defined above for perception has a counterpart in action. The counterpart of a transducer is an actuator. An actuator applies a change to the state of the external world. An actuator interprets a command, $A_i(t)$, which we will define to be a time sampled, digitized signal. Thus we group the motor controller, power amplifier, motor and mechanical system together as an actuator. A command may be a scalar, a vector, or have any number of dimensions. Most actuators interpret velocity or position commands specified as a scalar quantized values.

Each actuator operates in its own coordinate space. It is often preferable to specify actions in coordinates which relate to the device or to the external world. We define a device controller as an interpreter which transforms commands from a “virtual” device to the real actuators. A device controller interprets a time sampled digitized signal, $D_i(t)$. A parameter may be a scalar or a vector and provides a reference signal for the device controller. A common example of a device controller is the Cartesian arm controller which is standard for most robot manipulators. Other examples include a vehicle controller for a robotic vehicle [16] and the a Cartesian head controller for a binocular head [14].

A composition of parameters for device controllers and/or actuators forms an action space [4]. A command is a vector of parameters in an action space.

$$C(t) = \{D_1(t)...D_n(t)\}$$

2.3 Behaviors: From perception to action.

Using the above definitions, the behavior of a reactive process is defined as a transformation from a perceptual space to an action space.

$$\text{Reactive Process:} \quad C(t) \leftarrow B_i(P_k(t))$$

A large variety of techniques exist for defining such transformations. The classical approach is to use a PID controller. A modern control theory approach involves applying a controller based on a Kalman filter, lattice filter, of alpha-beta tracker. The use of fuzzy control is rapidly gaining popularity [25]. A large number of systems have recently been built using various artificial neural network approaches such as ART [8] and back propagation [23]. By defining reactive visual processes using virtual sensors, it becomes possible for such processes to exploit local models of the environment. Thus it is possible to take advantage of the reduction in complexity made possible by clever use of intermediate representations.

2.4 Predictions: From action to perception

In order to select the appropriate action, it is useful for a supervisory system to be able to predict the effect that an action will have on the external world. Since the system can not directly know the world state, it must perceive it through its transducers and virtual sensors. Thus predicting the effect of an action is equivalent to predicting the change in a perception from an action. We define a prediction as a transformation from an action space to perceptual space.

$$P_k(t+\Delta T) \leftarrow P_i(C(t))$$

As with a reactive visual process, a prediction may be defined by any number of techniques.

3 Selection and Control of Visual Processes

The subsumption architecture [5] posits the use of a simple hierarchy of processes using inhibition as a control mechanism. Such a mechanism assumes that the tasks of the system do not change. Efforts to construct such systems with more than a few "behaviours" soon leads to problems of which process should inhibit which and when. As an alternative, we propose to construct systems with a large repertoire of possible reactive visual processes (or behaviours or modes or controllers) and to use a supervisory controller to select the appropriate process based on current circumstances and goals.

3.1 Supervisory Control of Reactive Processes

The supervisory controller and its relation to the repertoire of visual processes is illustrated in figure 1. This figure shows a set of possible processes (B_1 through B_6) set up to receive their perceptual data from a set of virtual sensors and to produce commands for a set of device controllers. The currently active processes (shown as dark ellipses) are selected by the supervisory controller based on perceptual signals and current goals. Any conflicts in the commands issued by the processes are resolved by the device controllers.

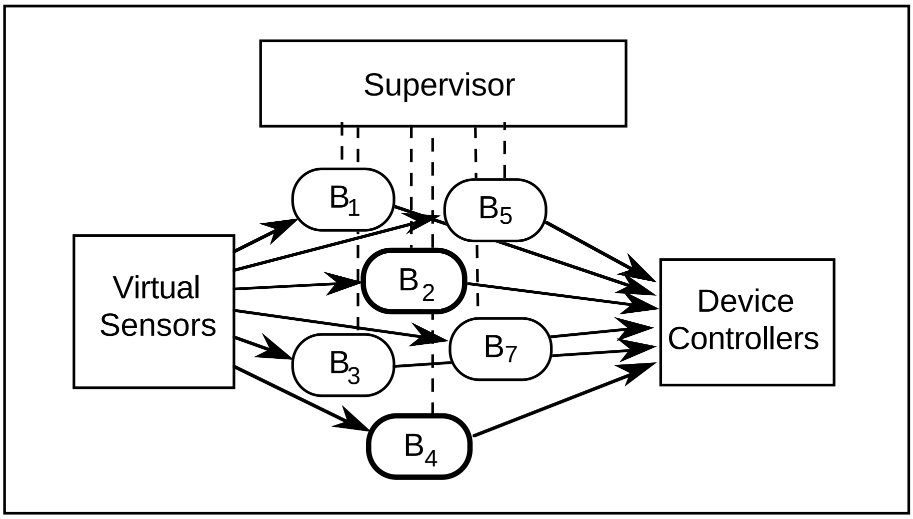

Figure 1 . A Supervisory controller selects and controls the sequencing of perceptual processes (shown as B for Behaviours). Multiple processes can be active at the same time. Arrows indicate flow of data, dashed lines indicate control, the highlighted ellipses are currently active.

A number of techniques are available to organise the supervisory controller. The most natural of these appears to be to organise the processes as a network of states, where each state corresponds to a set of reactive processes with associated control parameters. For each state, a set of possible next states can be selected based on both the current goals (or sub-goals) and on perceptual signals. Multiples states can be active at the same time, and transitions to states can be conditioned on unexpected events, such as detection of an impending collision, or the presence of a human master.

A state network approach provides a number of advantages.

- Networks can be abstracted by collapsing sub-networks into "super-states" to form a more abstract network. In this way, a system can reason hierarchically about its actions, reducing complexity.
- The same visual process (or sets of visual processes) with different parameters can be represented by different states.
- Formal methods exist to prove properties about such state transition networks.

Such an approach also makes it possible for planning techniques to be used in the design of state transitions networks, and provides an approach for control of plan execution. In this way, a mission may be specified as a sequence of tasks to be accomplished [12]. Each task can be translated into sub-goals expressed in terms of desired world state. The system can then select a sequence of reactive processes which may be applied at the current world state to transform the world to the desired state.

This approach also opens new problems. One such problem is the transition between reactive processes. It is relatively easy to construct pairs of reactive processes which drive the system back and forth between a transition and thus generate an oscillation. Even when there is no oscillation, care must be taken at the transition between reactive processes to avoid [18].

3.2 Selection and Sequencing by Signals

The Supervisory control problem for reactive processes can be expressed as selection and sequencing. Selection is the process of determining which reactive processes can next be executed. Sequencing determines when to make the transition to the next process. From the point of view of the reactive process, both selection and sequencing are controlled by signals. A signal triggers a change of reactive process. The value of the signal serves to select the next process, while the time of arrival of the signal serves to determine when the transition occurs.

We distinguish two kinds of signals: command signals and perceptual signals. Command signals flow from the supervisory controller to the reactive processes. These may be divided into two sub-classes: unconditional commands and conditional commands. An unconditional command orders an immediate transition to the new reactive process. A conditional command enables a transition to a new reactive process at the reception of an appropriate perceptual signal. In this way the delay in communications between the supervisor and the reactive controller can be avoided in the actual transition. A set of conditional commands can enable a set of possible reflex-level reactions to uncontrollable events. The state transition, whether conditional or unconditional, should be accompanied by an acknowledgement to the supervisor.

Perceptual signals are generated by a form of reactive process and are used to change the current set of reactive processes. Perceptual signals can be used to trigger the transition of reactive processes from external events with a minimum of delay. They can also be used as watch-dogs which enable the system to quickly react to uncontrollable events.

4 Example: A System for Detection, Fixation and Tracking

To illustrate our approach, we describe a minimal system designed as a composition of four reactive visual processes for detection, fixation and tracking. This example illustrates how a system composed of reactive visual processes can be designed to attend to dynamic events, including events which occur unexpectedly.

4.1 Attending to motion and while watching for faces.

The state transition network which describes the demonstration is shown in figure 2. The basic task of our demonstration is to watch for motion, and when motion is detected, to maintain a 3D fixation on the thing that moved. In its initial state, the system is looking for motion within the binocular visual field. If motion is detected by either camera, the system attempts to fixate the center of gravity of the motion using both cameras. Such fixation is servoed in 2D but provides an estimate of the fixation point in 3D head centered coordinates. When a region of motion is centered in both visual fields, the system switches to a mode in which a correlation tracker is used to hold the 3D fixation point on the object which is found in the center of the image. If fixation on the object is broken, the system reverts to the motion detection state. A face detection process operates in parallel with these processes. If a face is detected in either visual field, the system switches its fixation to the face. If the face correlation is lost, the system reverts to correlation tracking on the "form" found where the face was present (usually the human's head). In this way, a person can turn his back to the robot and walk away and the robot will follow.

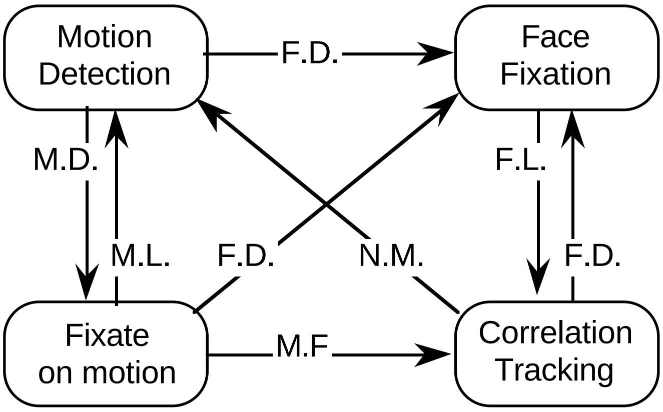

Figure 2. The State transition network for the demonstration.

4.2 Implementation in the SAVA III Test-bed

This example uses the SAVA III distributed vision test-bed [15]. SAVA III provides an infrastructure for experiments in continuously operating vision. The SAVA system is composed of a number of individual modules connected by a message passing facility implemented using sockets. Each SAVA module is constructed within an interpreter (CLIPS 5.1) which provides a lisp-like syntax for functions, rules and objects. This interpreter acts as a scheduler, a message interpreter and an interpreter for rule-based "demons".

Each module contains a collection of procedures which concern a data structure. A small set of rules provide a scheduler which calls the selected procedures in a cyclic manner. Because the scheduler is interpreted, the set of procedures and their parameters can be changed dynamically. Between each procedure call, the interpreter reads and interprets any messages recieved from the other modules. Such messages are typically used to interrogate local data structures, or to define the processing within a module. The SAVA modules send and receive messages encoded as ASCII strings using a mail box facility. The first word of each message is a function which is interpreted by the CLIPS interpreter using lisp-like "eval" function. The "build" command in CLIPS makes it possible to define a new message type with a message from another module. Using an interpreter for control and communications between modules has greatly accelerated experiments in control of perception.

The robotics part of the SAVA III system includes a binocular head mounted on a 6-axis manipulator, itself mounted on a mobile platform [13]. An image acquisition and processing module uses special purpose hardware to acquire synchronised stereo images and compute a half-octave binomial pyramid [10]. SAVA III provides independent modules as distributed processes for fixation control, navigation, image acquisition and processing, image description, 3D modeling, and system supervision. This particular demonstration uses only a subset of the available SAVA III modules. The system has been able to exploit existing capabilities for 3D fixation, for redundant control of the head and vehicle, and for reflex level control of focus, aperture and vergence.

Figure 3 shows the configuration of modules which are used for this demonstration. Two synchronised stereo cameras are connected to a module for image acquisition and processing. This module responds to requests from the system supervisor and from the fixation control module. The fixation control module contains state variables for the current 3D fixation point, and for 2D fixation points for each camera. Messages from the supervisor set the desired value for these fixations. Fixation control interprets the fixation command to generate commands to the 10-degree of freedom head-body system. If maintaining fixation requires movement by the vehicle, fixation control can send messages to the navigation system to perform the necessary movements. A synchronization module provides a global time reference so that all data and commands can be time-stamped. This time stamp is used to produce timing diagrams to illustrate the execution speeds of the modules as their processes change.

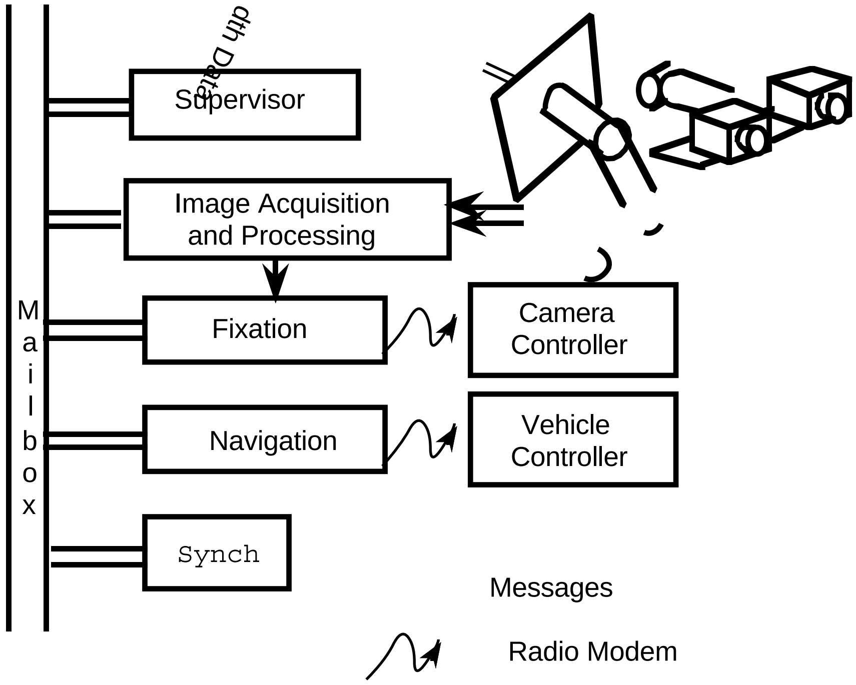

Figure 3 Configuration of demonstration system within the SAVA III Vision test-bed.

4.3 Vocabulary of Visual Processes

The visual processes which make up this demonstration include motion detection, fixation, face detection and tracking. Each of these processes depends on an image processing procedure executed by the image acquisisition and processing module, and commanded by the fixation module. For each process we identify the virtual sensors and the device level commands that are generated, and describe the transformation from perceptual space to action space

Motion Detection

The motion detection process is based on the energy in a temporal derivative of the images. The process relies on virtual sensor values:

Virtual Sensors:

M: The sum of squared difference of successive images

C_x, C_y: The bary-center of squared difference image.

The action space for this process is composed of the pan and tilt directions for camera:

Action Space:

α, ϕ Pan and tilt angle to bring bary-center of motion to the center of image.

The motion detection process is implemented as a sum of squared difference of successive images followed by a calculation of the bary-center of local energy. The input images are selected from one of the levels of a binomial pyramid [10]. The pyramid an important reduction in communication and computation. An resampled image size of 64 x 64, correponding to smoothing by a binomial filter ($\sigma = 4$) is used for the experiments described below. A difference image is computed from the previous image and then squared. If the sum of the squared difference is below a threshold, the process signals no motion. Otherwise, the barycenter is computed for the squared difference image. The row and column values of the barycenter, and the sum of squared difference constitute the virtual sensor for this first reactive behaviour.

The pan and tilt values are specified to the fixation controller. The fixation controller uses the sum of the pan angles to set the head orientation ϕ_4 and the difference to set the vergence angles of the cameras. When the head orientation reaches a limit, the system uses other axes, and the vehicle to turn the head towards the fixation point [14]. Figure 4 shows examples of images, the difference image, and the detected barycenter.

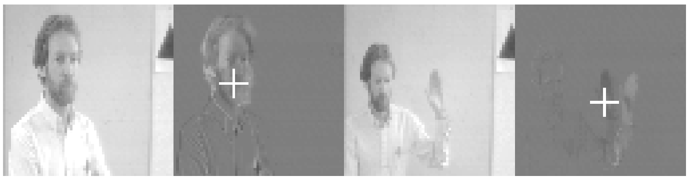

Figure 4. Motion detection. Two images and the difference with the previous image . The cross indicates the barycenter where motion was detected.

Fixation on Motion The motion fixation process operates by normalized correlation of a small template, typically 8 by 8 or 16 by 16. The template is registered at the barycenter of an image when motion is detected. The template is correlated with both the right and left images to indicate the pan and tilt angle for the left and right cameras. The area of the window over which the correlation is performed is large so that the moving pattern can be found in both cameras so that stereo convergence can be established. The virtual sensors are thus the position in each image at which the best normalized correlation is found and the normalized correlation (sum of squared difference) values.

Virtual Sensors:

D_r, D_l: Best normalized correlation scores for left and right images.

x_r, y_r: Image position of best correlation score in right image.

x_l, y_l: Image position of best correlation score in left image.

The action space is the pan and tilt angle required to bring the best correlation positions to the center of the left and right images.

Action Space:

α_r, ϕ_r Pan and tilt angle to bring correlation to center of right image.

α_l, ϕ_l Pan and tilt angle to bring correlation to center of left image.

Conversion of the pan and tilt angles to a 3D fixation point is performed by the fixation control module.

Tracking Fixation: Once the form is centered in both images, it is possible to reduce the search region, resulting in a gain in processing speed. The system moves to a tracking behaviour in which the virtual sensors are of the same nature as motion fixation, except that the search region is much smaller and at higher resolution. We have recently begun experiments in which the search region is a parameter of the delay since the last cycle of fixation.

Face Detection Face Detection is a back-ground signal detection process, programmed as a demon. Face detection operates by normalized correlation of a small (16 by 16) average face image with several levels of the Gaussian pyramid. The average face image has been formed by acquiring a number of face images of laboratory members with a neutral background, normalising their position and orientation, and then computing an average image. Detection of a face constitutes a perceptual signal which moves the system to the face fixation process. In face fixation, the virtual sensors are the row and column positions of the best correlation of the average face in the left and right images, and the sum of squared difference between the face window and each image at this best correlation value.

Virtual Sensors:

D_{fr}, D_{fl}: Best normalized correlation scores for left and right images.

x_{fr}, y_{fr}: Image position of best correlation score in right image.

x_{fl}, y_{fl}: Image position of best correlation score in left image.

The action space is the pan and tilt angle required to bring the face position to the center of each image.

Action Space: α, ϕ Pan and tilt angle to bring face to center of image.

The set of perceptual signals and their definitions are as follows.

F.D. The face detection signal is given by a threshold on normalized correlation (SSD) with the average face.

M.D. Motion detected is signaled by the sum of the squared temporal difference greater than a threshold.

M.F. The motion fixated signal is triggered when the motion field is within 16 pixels of the center of both images.

F.L. A face lost signal is detected when the best normalized correlation (sum of squared differences) of the average face with both images falls above a threshold.

N.M. The no motion signal is triggered when the tracked window has not moved by more than n pixels in last m images

M.L. A motion lost signal occurs in motion fixation if the motion signal (sum of squared temporal difference) falls above a threshold.

4.4 Systems Execution

Timing diagrams have proven to be a useful tool for debugging visual behaviours. Each module contains a synchronised clock. At the start of each cycle, the module retrieves the time that has elapsed since the start of the last cycle. This value is appended to a list which is output to a file at the end of execution.

Figure 4 illustrates timing in the image acquisition and processing module during a typical tracking session using 64 by 64 images. The vertical axis is cycle time measured in milliseconds, while the horizontal axis is the cycle number. The motion detection demon is invoked near cycle number 17, causing the cycle time to rise from 200 milliseconds to 360 milliseconds. At cycle number 101, the demon detected motion and sent a signal to engage the fixation process. During fixation, the cycle times oscilate between 500 milliseconds and 200 milliseconds. The 500 milliscond cycle occurs when a command has been received to compute correlation with the template in both images. A 200 milli-second cycle occurs when no command is received, and only image acquisision and pyramid computatio are performed. At cycle 111, fixation is achieved and the tracking demon is invoked. During tracking the module requires around 400 milliseconds to determine the correlation. At cycle number 276, tracking was lost, and the motion detection demon resumed operation.

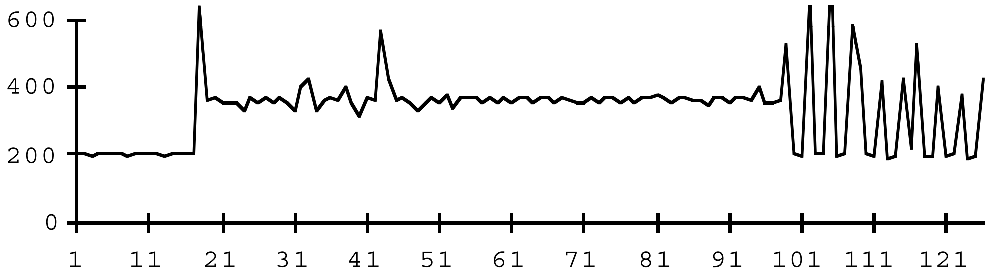

Figure 4. Timing diagram for image acquisistion and processing from a typical tracking session. Vertical axis is milli-seconds. Horizontal axis is cycle number.

5 Conclusions

Vision systems which can not dynamically control acquisition and processing are limited to a small number of task in a fixed environment. In order to integrate more than a few visual behaviours, an approach is required which permits data and processing to be dynamically selected in response to system goals and external events. This paper presents such an approach based on the concept of reactive vision processes.

The use of reactive processes for robotics has generally been restricted to transformations from sensor signals to motor commands. In order to overcome limitations imposed by the complexity of perception we define the reactive transformation on virtual sensors which may include local modeling systems, long term memory and even system goals. Commands are generated to device level controllers which integrate such commands with proprioceptive signals, and resolve contradictory commands. Formulating a continuously operating vision system in terms of such reactive processes permits the system to be scalable and to adapt to a changing and unpredictable environment.

This new approach to vision systems opens new problems, including techniques for

- Analysing the stability of compositions of reactive transformations.
- Learning reactive controllers and prediction functions for reactive processes.
- Learning to detect perceptual signals which are relevant to control.
- Learning the composition of visual processes to form perceptual "skills".

The tight coupling of vision and action and the introduction of learnable techniques may provide the keys to bringing computer vision out of the laboratory and into every-day use.

Bibliography

[1] J. Aloimonos, I. Weiss and A. Bandopadhay, "Active Vision", International Journal on Computer Vision, pp. 333-356, 1987.

[2] R. Bajcsy, "Active Perception", Proceedings of the IEEE , Vol 76, No 8, pp. 996-1006, August 1988.

[3] D. Ballard, "Animate Vision", Artificial Intelligence, Vol 48, No. 1, pp. 1-27, February 1991.

[4] A. G. Barto, "An approach to Learning Control Surfaces by Connectionist Systems", in M. Arbib and A. Hanson, Vision, Brain and Cooperative Computation, MIT Press, Cambridge MA 1987.

[5] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, RA-2(1) March 1986.

[6] C. Brown, "Prediction and Cooperation in Gaze Control", Biological Cybernetics 63, 1990.

[7] K. Brunnström, "Active Exploration of Static Scenes", Doctoral Dissertation, KTH - Royal School of Technology, Stockholm Sweden, 1993.

[8] Carpenter G. A. “Neural network models for pattern recognition and associate memory”, Neural Networks, Vol 2, 1989.

[9] D. Chapman and L. P. Kaelbling, "Learning from Delayed Reinforcement in a Complex Domain", Proc. of the IJCAI, 1991.

[10] A. Chehikian and J. L. Crowley, "Fast Computation of Optimal Semi-Octave Pyramids", 7th S.C.I.A., Aalborg, August 1991.

[11] Robot Learning, Edited by J. Connell and S. Mahadevan, Kluwer Academic Publishers, Boston, 1993.

[12] Crowley, J. L., "Coordination of Action and Perception in a Surveillance Robot", IEEE Expert, Vol 2(4), pp 32-43 Winter 1987, (Also appeared in IJCAI-87).

[13] Crowley, J. L. "Towards Continuously Operating Integrated Vision Systems for Robotics Applications", SCIA-91, Seventh Scandinavian Conference on Image Analysis, Aalborg, August 91.

[14] Crowley, J. L., P. Bobet and M. Mesrabi, “Camera Control for a Active Camera Head”, Pattern Recognition and Artificial Intelligence, Vol 7, No. 1, January 1993.

[15] Crowley, J. L.and H. I. Christensen, Vision as Process, Springer Verlag Basic Research Series, to appear 1993.

[16] Crowley, J. L. and Patrick Reignier, “Asynchronous Control of Rotation and Translation for a Robot Vehicle”, Robotics and Autonmous Systems, Vol 10, No. 1, January 1993.

[17] Eklundh, J. O. and K.Pahlavan, “A head-eye system: Analysis and Design.”, CVGIP, 56:1, 41-56

[18] J. A. Coelho and R. A. Grupen, “Constructing Effective Multifingered Grasp Controllers”, Submitted to the 1994 IEEE Conf. on Robotics and Automation, 1994.

[19] L. P. Kaelbling Learning in Embedded Systems, MIT Press, Cambridge Mass, 1993.

[20] J. Kosecka and R. Bajcsy, “Discrete Event Systems for Autonomous Mobile Agents”, Intelligent Robotic Systems, ‘93, Zakopane, 1993 (also to appear in Robotics and Autonomous Systems, 12(3) march 94.

[21] Krotkov, E., "Focusing", International Journal of Computer Vision, 1, p223-237(1987).

[22] Krotkov, E., Henriksen, K. and Kories, R., “Stereo Ranging from Verging Cameras”, IEEE Trans on PAMI, Vol 12, No. 12, pp. 1200-1205, December 1990.

[23] D. A. Pomerlau, "Neural Network Based Autonomous Navigation", in Vision and Navigation, C. Thorpe (ed)., Kluwer Academic Publishers, Boston, 1990.

[24] P. J. Ramadge and W. M Wonham, “The Control of Discrete Event Systems”, Proceedings of the IEEE, 77(1), January 1989.

[25] P. Reignier, “Fuzzy Logic Techniques for Mobile Robot Obstacle Avoidance”, Intelligent Robotic Systems, ‘93, Zakopane, 1993 (also in Robotics and Autonomous Systems, 12(3) march 94.

[26] K. Souccar, M. Huber, and J. A. Coelho, "Sequencing Contollers - Experiments in Auronomous Reaching and Grasping", 1994 IEEE Conference on Robotics and Automation, May 1994.

[27] R. S. Sutton, "Integrated Architectures for Learning, Planning and Reacting Based on Approximating Dynamic Programming", in Proceedings of the 7th Int. Conf. on Machine Learning, June 1990.

[28] J.K. Tsotsos, “Representational Axes and Temporal Co-operative Processes, In: Vision”, Brain and Co-operative Computation, (Eds.) M.A. Arbib & A.R. Hanson, MIT Press, Cambridge, Mass, pp. 361-418, 1987.

[29] Westelius, C. J., H. Knutsson, and G. H. Granlund, "Focus of Attention Control", SCIA-91, Seventh Scandinavian Conference on Image Analysis, Aalborg, Aug. 91.